WHAT IS CLAIMED IS:

1. A method for generating a color value for a pixel from geometry data, the method comprising:

selecting a first plurality of shading locations and a second plurality of depth locations for the pixel, the second plurality being larger in number than the first plurality, each of the second plurality of depth locations being associated with one of the shading locations;

under control of a graphics processing subsystem, generating a plurality of hybrid sampled data points equal in number to the second plurality of depth locations, wherein the act of generating includes:

computing a shading value at each of the first plurality of shading locations and a depth value at each of the second plurality of depth locations; and

storing one of the depth values and the associated shading value as one of the hybrid sampled data points; and

computing an antialiased color value for the pixel using the hybrid sampled data points.

2. The method of claim 1 wherein the act of generating the plurality of hybrid sampled data points includes:

performing at least two multisampling operations on the pixel,

wherein each multisampling operation uses a different one of the shading locations and a different subset of the depth locations and generates a different subset of the plurality of hybrid sampled data points.

3. The method of claim 2 wherein the subset of the hybrid sampled data points generated by each multisampling operation is stored in a corresponding one of a plurality of target buffers.

4. The method of claim 2 wherein the subset of the hybrid sampled data points generated by each multisampling operation is accumulated in an accumulation buffer.

5. The method of claim 1 wherein each of the depth locations is inside the pixel.

6. The method of claim 1 wherein each of the shading locations is inside the pixel.

7. The method of claim 1 wherein the geometry data includes a primitive, the method further comprising, prior to storing one of the depth values and the associated shading value, determining whether the primitive covers the depth location,

wherein the one of the depth value and the associated one of the shading value are not stored in the event that the primitive does not cover the depth location.

8. The method of claim 1 wherein the act of selecting the first plurality of shading locations and the second plurality of depth locations for the pixel includes:

segmenting a viewable area that includes the pixel into a number of sub-pixels, each sub-pixel having a size smaller than a size of the pixel,

wherein each sub-pixel includes one of the shading locations and a subset of the depth locations.

9. The method of claim 8 wherein associating each of the second plurality of depth locations with one of the shading locations includes:

associating each of the depth locations of a sub-pixel with the shading location of that sub-pixel.

10. The method of claim 8 wherein the pixel is divided into an integer number of sub-pixels.

11. The method of claim 8 wherein the act of segmenting the viewable area includes providing a multisampling rasterizer with a display resolution that is larger than a true display resolution.

12. The method of claim 10 wherein the pixel is divided into four sub-pixels arranged to form a quad.

13. The method of claim 1 wherein the act of selecting the first plurality of shading locations and the second plurality of depth locations for the pixel includes:

defining a multisampling pattern for the pixel, the multisampling pattern including one of the depth locations and at least two of the shading locations;

generating a plurality of iterations of the geometry data, wherein each iteration has a different offset relative to a boundary of the pixel; and

applying the multisampling pattern to each of the iterations of the geometry data.

14. The method of claim 13 wherein each of the offsets corresponds to an amount less than a pixel size.

15. The method of claim 14 wherein one of the offsets is equal to zero.

16. The method of claim 13 wherein the act of generating the plurality of iterations includes, for each iteration, setting a value of a viewport offset parameter corresponding to the offset of the iteration.

17. The method of claim 13 wherein generating the plurality of hybrid sampled data points includes:

storing the depth values and the associated shading value obtained from each iteration in a respective one of a plurality of buffers.

18. The method of claim 1 wherein the act of selecting the first plurality of shading locations and the second plurality of depth locations for the pixel includes:

defining a multisampling pattern for the pixel, the multisampling pattern including one of the depth locations and at least two of the shading locations;

defining a plurality of non-overlapping regions in an image coordinate space, each region including a virtual pixel corresponding to the pixel;

relocating the geometry data to a position within each of the regions, wherein the position of the relocated geometry data relative to a boundary of the region is shifted by an amount less than a pixel size; and

applying the multisampling pattern to each of the virtual pixels.

19. The method of claim 18 wherein one of the regions corresponds to a viewable area of the image coordinate space.

20. The method of claim 18 wherein for one of the regions, the amount less than a pixel size is zero.

21. The method of claim 18 wherein relocating the geometry data includes:

setting a value of a window offset parameter such that the geometry data is placed within one of the regions; and

setting a value of a viewport offset parameter corresponding to the shift by an amount less than a pixel size.

22. The method of claim 18 wherein the act of relocating the geometry data is performed by the graphics processing subsystem.

23. The method of claim 1 wherein the act of computing the color value for the pixel includes:

defining a texture map including a second plurality of texels corresponding to the hybrid sampled data points for the pixel;

fetching the second plurality of texels; and

computing a weighted average of the fetched texels, thereby determining the color value for the pixel.

24. The method of claim 1 wherein the act of computing the color value for the pixel includes:

defining a plurality of texture maps, each texture map including a plurality of texels corresponding to a subset of the hybrid sampled data points for the pixel;

for each of the plurality of texture maps:

fetching the plurality of texels from the texture map; and

blending the fetched texel values to generate an intermediate value; and

computing a weighted average of the intermediate value generated for each of the texture maps, thereby determining the color value for the pixel.

25. The method of claim 1 wherein the act of computing the color value for the pixel is performed during a scanout operation that provides downfiltered color data to a display device.

26. The method of claim 1 wherein the acts of generating the plurality of hybrid sampled data points and computing the color value for the pixel are performed in a single rendering pass.

27. The method of claim 1 wherein the number of shading locations and the number of depth locations are determined based on one or more configurable parameters.

28. A system for generating a color value for a pixel from geometry data, the system comprising:

a multisampling rasterizer configured to receive the geometry data and perform a multisampling operation on the pixel, the multisampling operation generating a plurality of depth values at a plurality of depth locations for the pixel and one shading value, the shading value being associated with each of the plurality of depth locations;

control logic configured to use the multisampling rasterizer to perform a plurality of multisampling operations on the pixel; and

a downfiltering unit configured to combine the shading values generated during the plurality of multisampling operations, thereby generating a color value for the pixel.

29. The system of claim 28 wherein the control logic is further configured to select different depth locations for each of the plurality of multisampling operations.

30. The system of claim 28 wherein the control logic is further configured to change a screen location of the geometry data such that the multisampling rasterizer uses different depth locations for each of the plurality of multisampling operations.

31. The system of claim 28 wherein the control logic is further configured to use the multisampling rasterizer to perform a multisampling operation on the geometry data for each of a plurality of sub-pixels at different locations within the pixel.

32. The system of claim 31 wherein the multisampling rasterizer is instructed to use a display resolution larger than a true display resolution.

33. The system of claim 28 wherein the control logic is further configured to store the geometry data and to supply the geometry data to the multisampling rasterizer multiple times in succession.

34. The system of claim 28 wherein the control logic is further configured to relocate the geometry data in each of a plurality of non-overlapping regions and to instruct

the multisampling rasterizer to perform a multisampling operation on a virtual pixel in each region.

35. The system of claim 34 further comprising a buffer having a plurality of non-overlapping regions, wherein multisampled pixel data from each of the non-overlapping region is stored in a respective one of the non-overlapping regions.

36. The system of claim 34 further comprising a plurality of buffers, wherein multisampled pixel data from each of the non-overlapping regions is stored in a respective one of the plurality of buffers.

37. The system of claim 28 further comprising:

a frame buffer for storing the shading value at each depth location,

wherein the downfiltering unit is further configured to read the shading values from the frame buffer.

38. The system of claim 37 wherein the downfiltering unit includes:

a texture processing unit configured to fetch at least one of the shading values from the frame buffer as a texel and to generate an intermediate value from the texel; and

a shader configured to blend the intermediate values, thereby generating the color value for the pixel.

39. The system of claim 38 wherein:

the texture processing unit is further configured to fetch all of the shading values for the pixel from the frame buffer and to provide each fetched shading value as an intermediate value.

40. The system of claim 38 wherein:

the texture processing unit is further configured to fetch a plurality of subsets of the shading values for the pixel from the frame buffer and to blend each subset of the shading values, thereby generating a plurality of intermediate values.

41. An apparatus for generating a color value for a pixel from geometry data, the apparatus comprising:

a graphics processor including:

a multisampling rasterizer configured to receive the geometry data and perform a multisampling operation on the pixel, the multisampling operation generating a plurality of depth values at a plurality of depth locations for the pixel and one shading value, the shading value being associated with each of the plurality of depth locations;

control logic configured to use the multisampling rasterizer to perform a plurality of multisampling operations on the pixel; and

a downfiltering unit configured to combine the shading values generated during the plurality of multisampling operations, thereby generating a color value for the pixel;

a frame buffer configured to store the shading values generated during the plurality of multisampling operations; and

a downfiltering unit configured to combine the shading values stored in the frame buffer, thereby generating a color value for the pixel.

42. The apparatus of claim 41 further comprising:

a graphics driver module configured to communicate with the graphics processor and to configure a parameter for the plurality of multisampling operations.

43. The apparatus of claim 42 wherein the parameter determines a number of multisampling operations to be performed.

44. The apparatus of claim 42 wherein the parameter determines a number of depth locations to be used during each of the multisampling operation.

45. The apparatus of claim 42 wherein the graphics driver module includes an application program interface for configuring the parameter.

46. The apparatus of claim 42 wherein the graphics driver module includes a user interface for configuring the parameter.

47. The apparatus of claim 42 wherein the graphics driver module is further configured to detect a property of an application program and to configure the parameter based at least in part on the detected property.